UNIVERSITAT POLITÈCNICA DE CATALUNYA
(UPC) – BarcelonaTech

FACULTAT D'INFORMÀTICA DE BARCELONA (FIB)

MASTER IN INNOVATION AND RESEARCH IN INFORMATICS - DATA SCIENCE

# Data analysis of socio-economic and financial factors from a public world-wide source

Director/a:
Prof. Marta Arias

Autor:
Santiago Calvo Fantova

Academic year 19/20

Date - 27/01/2020

## 1. Abstract[1]

Many socio-economic studies are nowadays trying to accomplish a complete description of how the different elements of our society and world are connected. This work is an attempt to build an architecture that provides an explanation of the connections and impacts that exist among the different indicators (for now on also mentioned as sectors) of a state or population (Agriculture, Climate Change, Economy & Growth, Energy & Mining, Education, Health, Poverty, Science & Technology, Social Development, and others).

We will focus our effort in the research of, not only the correlations that may exist between these indicators and thought the different countries analyzed, but also the causality that relates them. With causality (we will deploy a Bayesian Network architecture for each country to accomplish this task), we will be able to describe the impact and influence that one indicator may have in the others. This could lead to an accurate, powerful and global knowledge of the functioning of our world and each single country in particular, along with a vision of the dependencies between the different indicators that describe a country.

Finally, we will also propose a clustering model where each individual will be a representation of the Bayesian Network obtained for each country. With this model, we will provide *N* aggrupation of countries with their Bayesian Network representation for each one, which will give us a global vision of the functioning of our world represented by the causal relationships between the different indicators that can be found in our countries or populations.

## 2. Introduction, motivation and goals

The idea of this project born with the following questions: Given the huge, historical, complex and diverse amount of data that exist nowadays for one country or population; Is it possible to extract an accurate description of the functioning of a country from it?; Is it possible to create models that group the different countries in some clusters?; Can we create a global description of the functioning of our world?.

We are aware that there exist many studies that are focused on the analysis of the different correlations between the diverse data variables that can be found for one country. But we wanted to go one step further and focus the analysis on not only the possible correlations that could exist between data variables but also the causal relationships that might exist among them. This is one of the key points of our project, the fact that we focus on the search of causality instead of correlation.

The previous questions arise on us the necessity of developing an architecture that could try to give an answer to them. We first found the source of our project, we were looking for a good-quality statistical data of countries indicators and we found "The World Bank"[1]. The World Bank Group is one of the world's largest sources of funding and knowledge for developing countries. One dataset per country can be downloaded, it contains data for the last 60 years (from 1960) and for more than 1500 variables grouped in 20 different sectors (Agriculture, Climate Change, Economy & Growth, Energy & Mining, Education, Health, Poverty, Science & Technology, Social Development, and others). We downloaded the data for the top 100 countries in the world with the highest GDP index. Then, our goal was to use

[1] The sectors and countries will be sometimes referenced with a shortening, to understand them we provide a dictionary in another attached document.

this huge dataset to develop a model to study the causal relationships between these sectors and try to group these causal models (one for each country) in different sets so that we can clusterize the total set of countries in some different groups. And finally, provide a global model for each of those groups or clusters.

## 3. State of the art

In this section, we will talk about the state of the art of the different methods and algorithms that we have used in our project and we will give a description and explanation of all them.

Each sub-section is referenced to the source and authors of the corresponding books and articles from where we have gathered this theoretical information.

### a. Dimensionality reduction algorithms [2]

In the modern age of technology, increasing amounts of data are produced and collected. In machine learning, however, too much data can be a bad thing. At a certain point, more features or dimensions can decrease a model's accuracy since there is more data that needs to be generalized, this is known as the curse of dimensionality.

Dimensionality reduction is a way to reduce the complexity of a model and avoid overfitting. There are two main categories of dimensionality reduction: feature selection and feature extraction. Via feature selection, we select a subset of the original features, whereas in feature extraction, we derive information from the feature set to construct a new feature subspace.

We have considered two different dimensionality reduction methods, the well-known and linear method Principal Component Analysis (PCA) and a most recent non-linear methodology that consist of the use of a neural network for feature extraction (known as Autoencoders).

#### i. PCA [2]

Feature sets can be more compact than the data they represent. Dimension reduction provides compact representations for storage, transmission, and classification. Dimension reduction algorithms operate by identifying and eliminating statistical redundancies in the data. The optimal linear technique for dimension reduction is principal component analysis (PCA). PCA performs dimension reduction by projecting the original *n-dimensional* data onto the $m < n$ dimensional linear subspace spanned by the leading eigenvectors of the data's covariance matrix. Thus, PCA builds a global linear model of the data (an *m-dimensional* hyperplane). Since PCA is sensitive only to correlations, it fails to detect higher-order statistical redundancies. One expects non-linear techniques to provide better performance, which is more compact representations with lower distortion.

This unsupervised linear transformation technique is widely used across different fields, most prominently for feature extraction and dimensionality reduction.

PCA helps us to identify patterns in data based on the correlation between features. Briefly, PCA aims to find the directions of maximum variance in high-dimensional data

and projects it onto a new subspace with equal or fewer dimensions than the original one.

The orthogonal axes (principal components) of the new subspace can be interpreted as the directions of maximum variance given the constraint that the new feature axes are orthogonal to each other, as illustrated in the *Figure 1*.

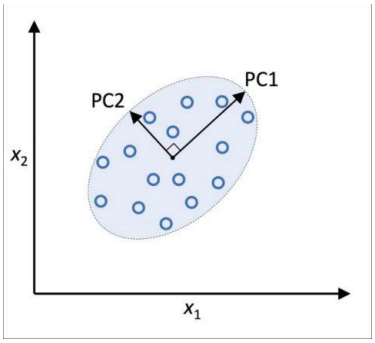


*Figure 1. PCA representation in two spaces or dimensions example.*

In the preceding figure, *x1* and *x2* are the original feature axes, and *PC1* and *PC2* are the principal components.

If we use PCA for dimensionality reduction, we construct a $d\ x\ k$ -dimensional transformation matrix *W* that allows us to map a sample vector *x* onto a new *k–dimensional* feature subspace that has fewer dimensions than the original *d–dimensional* feature space:

$$x = [x_1, x_2, \dots, x_d], \quad x \in \mathbb{R}^d$$

$$xW, \quad W \in \mathbb{R}^{dxk}$$

$$z = [z_1, z_2, \dots, z_k], \quad z \in \mathbb{R}^d$$

As a result of transforming the original *d-dimensional* data onto this new *k-dimensional* subspace (typically $k \ll d$), the first principal component will have the largest possible variance, and all consequent principal components will have the largest variance given the constraint that these components are uncorrelated (orthogonal) to the other principal components. Even if the input features are correlated, the resulting principal components will be mutually orthogonal (uncorrelated).

It is important to point out that the PCA directions are highly sensitive to data scaling, thus, we need to standardize the features prior to PCA if the features were measured on different scales and we want to assign equal importance to all features.

Therefore, in summary, PCA algorithm could be described in the following simple steps:

1. Standardize the *d*-dimensional dataset.
2. Construct the covariance matrix.
3. Decompose the covariance matrix into its eigenvectors and eigenvalues.

4. Sort the eigenvalues by decreasing order to rank the corresponding eigenvectors.
5. Select *k* eigenvectors which correspond to the *k* largest eigenvalues, where *k* is the dimensionality of the new feature subspace ($k \leq d$).
6. Construct a projection matrix ***W*** from the "top" *k* eigenvectors.
7. Transform the *d*-dimensional input dataset ***X*** using the projection matrix ***W*** to obtain the new *k*-dimensional feature subspace.

This is, therefore, how we get the features extracted for dimensionality reduction.

### ii. Autoencoders [3][4]

An autoencoder is an unsupervised artificial neural network that learns how to efficiently compress and encode data then learns how to reconstruct the data back from the reduced encoded representation to a representation that is as close to the original input as possible. Autoencoder, by design, reduces data dimensions by learning how to ignore the noise in the data.

In the *Figure 2* we display an illustrative example of an autoencoder where the source data is an image.

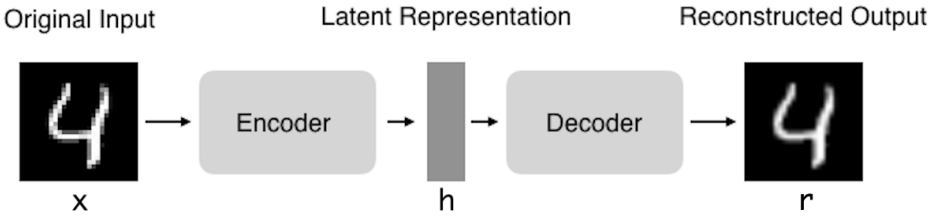


*Figure 2. Example of autoencoder compression for image data.*

Autoencoders are neural networks that aims to copy their inputs to their outputs. They work by compressing the input into a latent-space representation, and then reconstructing the output from this representation. This kind of network is composed of two parts:

- **Encoder**: This is the part of the network that compresses the input into a latent-space representation. It can be represented by an encoding function *h=f(x)*.
- **Decoder**: This part aims to reconstruct the input from the latent space representation. It can be represented by a decoding function *r=g(h)*.

The autoencoder as a whole can thus be described by the function *g(f(x)) = r* where we want *r* as close as the original input *x*.

- **Copying the input to the output:**

If the only purpose of autoencoders was to copy the input to the output, they would be useless. Indeed, we hope that, by training the autoencoder to copy the input to the output, the latent representation h will take on useful properties.
This can be achieved by creating constraints on the copying task. One way to obtain useful features from the autoencoder is to constrain *h* to have smaller dimensions than

*x*, in this case the autoencoder is called undercomplete. By training an undercomplete representation, we force the autoencoder to learn the most salient features of the training data. If the autoencoder is given too much capacity, it can learn to perform the copying task without extracting any useful information about the distribution of the data. This can also occur if the dimension of the latent representation is the same as the input, and in the overcomplete case, where the dimension of the latent representation is greater than the input. In these cases, even a linear encoder and linear decoder can learn to copy the input to the output without learning anything useful about the data distribution. Ideally, one could train any architecture of autoencoder successfully, choosing the code dimension and the capacity of the encoder and decoder based on the complexity of distribution to be modeled.

- **What are autoencoders used for?**

Today data denoising and dimensionality reduction for data visualization are considered as two main interesting practical applications of autoencoders. With appropriate dimensionality and sparsity constraints, autoencoders can learn data projections that are more interesting than PCA or other basic techniques.
Autoencoders are learned automatically from data examples. It means that it is easy to train specialized instances of the algorithm that will perform well on a specific type of input and that it does not require any new engineering, only the appropriate training data.
However, there are some fields where autoencoders have important limitations and they are not working very well, this is the case of image compression, for example. As the autoencoder is trained on a given set of data, it will achieve reasonable compression results on data similar to the training set used but will be poor general-purpose image compressors.

### b. Imputation methods [5]

Missing data is a common and exciting problem in statistical analysis and machine learning. They are necessary for evaluating data quality and can have different sources such as users not responding to questions in a recommender system, death of patients on treatment or non-compliance, errors in a database that describes the maintenance information of plant equipment, and so on.

**Missing Data Mechanism:**
To understand the importance of missing data, we need to identify the reasons for missing data occurrence. The first step is to understand the data and more importantly, the data collection process. This can lead to the possibility of reducing data collection errors. The nature or mechanism of missing data can be categorized into three major classes. These categories are based on the degree of relationship between the nature of the missing data and observed values:

- **Missing Completely at Random (MCAR):** This means that the nature of the missing data is not related to any of the variables, whether missing or observed. In this case, the missingness on the variable is completely unsystematic.
- **Missing at Random (MAR):** This means that the nature of the missing data is related to the observed data but not the missing data.

- **Missing Not at Random (MNAR):** This is also known as non-ignorable because the missingness mechanism cannot be ignored. They exist when the missing values are neither MCAR or MAR. The missing values on the variable are related to that of both the observed and unobserved variables.

The easiest way to assume a missing data mechanism from data is understanding the data collection process and use substantive scientific knowledge (critical in determining randomness in a missing data). The second method to understand the type of missing data mechanism is statistical testing. This method is mostly used when trying to figure out if the mechanism is either MAR or MCAR.

**Handling Missing Data:**
There are several methods used for treating missing data. Some of these methods started gaining a resurgence in the last decade because of their importance in clinical trials and biomedical studies. In addition, there are certain drawbacks associated with each of these methods when used for data mining and one needs to be careful to avoid bias or the under- or over-estimation of variability.

- **Mean, Median and Mode Imputation:**
  Using the measures of central tendency involves substituting the missing values with the mean or median for numerical variables and the mode for categorical variables. The major limitation of using this method is that it leads to biased estimates of the variances and covariance. The standard errors and test statistics can also be underestimated and overestimated respectively. This imputation technique works well with when the values are missing completely at random.

- **Imputation with Regression:**
  This is an imputation technique that uses information from the observed data to replace the missing values with predicted values from a regression model. The major drawback of using this method is that it reduces variability and overestimates the model fit and correlation coefficient.

- **k-Neareast Neighbor (kNN) Imputation:**
  For k-Nearest Neighbor imputation, the missing values are based on a kNN algorithm. These values are obtained by using similarity-based methods that rely on distance metrics (Euclidean distance, Jaccard similarity, Minkowski norm etc). They can be used to predict both discrete and continuous attributes. The main disadvantage of using kNN imputation is that it becomes time-consuming when analyzing large datasets because it searches for similar instances through all the dataset. Choosing the correct value for the number of neighbors (k) is also an important factor to consider when using kNN imputation.

- **Multiple Imputation using MICE (Multiple Imputation by Chained Equations):**
  Multiple imputation is a process where the missing values are filled multiple times to create “complete” datasets. Multiple imputation has many advantages over traditional single imputation methods. Multiple Imputation by Chained Equations (MICE) is an imputation method that works with the assumption that the missing data are Missing at Random (MAR). Recall that for MAR, the nature of the missing data is related to the observed data but not the missing data. The

MICE algorithm works by running multiple regression models and each missing value is modeled conditionally depending on the observed (non-missing) values.

### c. Bayesian Networks [6][7][8][9][10]

Bayesian networks are probabilistic models based on direct acyclic graphs. These models enable a direct representation of causal relations between variables. Their structure is ideal for combining prior knowledge, which often comes in causal form, and observed data.

These networks, also known as belief networks, belong to the family of probabilistic graphical models (GMs). These graphical structures are used to represent knowledge about an uncertain domain. In particular, each node in the graph represents a random variable, while the edges between the nodes represent probabilistic dependencies among the corresponding random variables. These conditional dependencies in the graph are often estimated by using known statistical and computational methods. Hence, BNs combine principles from graph theory, Probability Theory, computer science, and statistics.

GMs with undirected edges are generally called Markov random fields or Markov networks. These networks provide a simple definition of independence between any two distinct nodes based on the concept of a Markov blanket. Markov networks are popular in fields such as statistical physics and computer vision.

BNs correspond to another GM structure known as a directed acyclic graph (DAG) that is popular in the statistics, the machine learning, and the artificial intelligence societies. BNs are both mathematically rigorous and intuitively understandable. They enable an effective representation and computation of the joint probability distribution (JPD) over a set of random variables.

The structure of a DAG is defined by two sets: the set of nodes (vertices) and the set of directed edges. The nodes represent random variables and are drawn as circles labeled by the variable names. The edges represent direct dependence among the variables and are drawn by arrows between nodes. In particular, an edge from node *Xi* to node *Xj* represents a statistical dependence between the corresponding variables. Thus, the arrow indicates that a value taken by variable *Xj* depends on the value taken by variable *Xi*, or roughly speaking that variable *Xi* "influences" *Xj*. Node *Xi* is then referred to as a parent of *Xj* and, similarly, *Xj* is referred to as the child of *Xi*. An extension of these genealogical terms is often used to define the sets of "descendants"—the set of nodes that can be reached on a direct path from the node, or "ancestor" nodes—the set of nodes from which the node can be reached on a direct path. The structure of the acyclic graph guarantees that there is no node that can be its own ancestor or its own descendent. Such a condition is of vital importance to the factorization of the joint probability of a collection of nodes as seen below. Note that although the arrows represent direct causal connection between the variables, the reasoning process can operate on BNs by propagating information in any direction.

A BN reflects a simple conditional independence statement. Namely that each variable is independent of its non-descendants in the graph given the state of its parents. This property is used to reduce, sometimes significantly, the number of parameters that are

required to characterize the JPD of the variables. This reduction provides an efficient way to compute the posterior probabilities given the evidence.

In addition to the DAG structure, which is often considered as the "qualitative" part of the model, one needs to specify the "quantitative" parameters of the model. The parameters are described in a manner which is consistent with a Markovian property, where the conditional probability distribution (CPD) at each node depends only on its parents. For discrete random variables, this conditional probability is often represented by a table, listing the local probability that a child node takes on each of the feasible values—for each combination of values of its parents. The joint distribution of a collection of variables can be determined uniquely by these local conditional probability tables (CPTs).

Thus, we can define a Bayesian network *B* as an annotated acyclic graph that represents a JPD over a set of random variables *V*. The network is defined by a pair $B = \langle G, \Theta \rangle$, where *G* is the DAG whose nodes *X1, X2, …, Xn* represents random variables, and whose edges represent the direct dependencies between these variables. The graph *G* encodes independence assumptions, by which each variable *Xi* is independent of its non-descendants given its parents in *G*. The second component $\Theta$ denotes the set of parameters of the network. This set contains the parameter equation image for each realization *xi* of *Xi* conditioned on *i*, the set of parents of *Xi* in *G*. Accordingly, *B* defines a unique JPD over *V*, namely:

$$P_B(X_1, X_2, \dots, X_n) = \prod_{i=1}^{n} P_B(X_i|\pi_i) = \prod_{i=1}^{n} \theta_{X_i|\pi_i}$$

If *Xi* has no parents, its local probability distribution is said to be unconditional, otherwise it is conditional. If the variable represented by a node is observed, then the node is said to be an evidence node, otherwise the node is said to be hidden or latent.

- **Inference via BN:**

  Given a BN that specified the JPD in a factored form, one can evaluate all possible inference queries by marginalization, which is summing out over "irrelevant" variables. Two types of inference support are often considered: predictive support for node *Xi*, based on evidence nodes connected to *Xi* through its parent nodes (also called top-down reasoning), and diagnostic support for node *Xi*, based on evidence nodes connected to *Xi* through its children nodes (also called bottom-up reasoning). Such a support is formulated as follows:

$$P(C = T|A = T) = \frac{P(C = T, A = T)}{P(A = T)}$$

  Where

$$P(C = T, A = T) =$$
$$= \sum_{S,W,B,C \in \{T,F\}} P(C = T)P(S) \times P(W|C = T)P(B|S, C = T)P(A = T|B)$$

And

$$P(A = T) = \sum_{S,W,B,C \in \{T,F\}} P(C)P(S)P(W|C)P(B|S,C) \times P(A = T|B)$$

- **BN Learning:**

In many practical settings the BN is unknown and one needs to learn it from the data. This problem is known as the BN learning problem, which can be stated informally as follows: Given training data and prior information (like expert knowledge, casual relationships), estimate the graph topology (network structure) and the parameters of the JPD in the BN.

Learning the BN structure is considered a harder problem than learning the BN parameters. Moreover, another obstacle arises in situations of partial observability when nodes are hidden or when data is missing.

There are some different methods to accomplish this problem. One example is the one where the goal of learning is to find the values of the BN parameters (in each CPD) that maximize the log-likelihood of the training dataset. This dataset contains *m* cases that are often assumed independent. Given training dataset $\Sigma = \{\boldsymbol{x_1}, \ldots, \boldsymbol{x_m}\}$, where $\boldsymbol{x}_l = (x_{l1}, \ldots, x_{ln})^T$, and the parameter set $\Theta = (\vartheta_1, \ldots, \vartheta_n)$ where $\vartheta_i$ is the vector of parameters for the conditional distribution of variable $X_i$ (represented by one node in the graph), the log-likelihood of the training dataset is a sum of terms, one for each node:

$$logL(\theta|\Sigma) = \sum_m \sum_n logP\,(x_{li}|\pi_i, \theta_i)$$

The log-likelihood scoring function decomposes according to the graph structure. Hence, one can maximize the contribution to the log-likelihood of each node independently.

Another alternative is to assign a prior probability density function to each parameter vector and use the training data to compute the posterior parameter distribution and the Bayes estimates. To compensate for zero occurrences of some sequences in the training dataset, one can use appropriate (mixtures of) conjugate prior distributions. Such an approach results in a maximum a posteriori estimate and is also known as the equivalent sample size (ESS) method.

The solution that we chose for the task of structural learning of BNs in the space of DAGs, was the hill climbing algorithm which is considered to be the most used algorithm applied for this task. This learning approach is computationally efficient and, even though it does not guarantee an optimal result, many previous studies have shown that it obtains very good solutions. Hill climbing algorithms are particularly popular because of their good trade-off between computational demands and the quality of the models learned. Indeed, its success is due to its ease of implementation, efficiency and the quality of the obtained output, which is a locally optimal solution.

### d. Clustering algorithms [11][12]

Clustering is known as the process of grouping similar entities together. The goal of this unsupervised machine learning technique is to find similarities in the data point and group similar data points together.

There are two main clusterization algorithms which we will describe in this section, K-means and Hierarchical clustering.

- **K-means**

  The first step of this algorithm is creating, among our unlabeled observations, *c* new observations, randomly located, called centroids. The number of centroids will be representative of the number of output classes. Now, an iterative process will start, made of two steps. First, for each centroid, the algorithm finds the nearest points (in terms of distance that is usually computed as Euclidean distance) to that centroid, and assigns them to its category. Second, for each category (represented by one centroid), the algorithm computes the average of all the points which has been attributed to that class. The output of this computation will be the new centroid for that class.
  Every time the process is reiterated, some observations, initially classified together with one centroid, might be redirected to another one. Furthermore, after several reiterations, the change in centroids' location should be less and less important since the initial random centroids are converging to the real ones. This process ends when there is no more change in centroids' position.

- **How to decide the number of centroids?**

  Deciding the number of centroids is not an easy and straightforward task and it depends a lot in the goal of the clustering, the data itself and our own judgement. Still, we will give a brief description of some algorithms that will help us in the task of assessing the number of clusters:

  a) Elbow method: The elbow method is a heuristic method of interpretation and validation of consistency within cluster analysis designed to help find the appropriate number of clusters in a dataset. It is often ambiguous and not very reliable, and hence other approaches for determining the number of clusters such as the silhouette method are preferable.
  b) Silhouette method: is a measure of data consistency, the silhouette plot displays a measure of how close each point in one cluster is to points in the neighboring clusters.

- **Hierarchical Clustering**

  This algorithm can use two different techniques, Agglomerative and Divisive.

  Both algorithms are based on the same ground idea, but work in the opposite way. Being *K* the number of clusters (which can be set exactly like in K-means)

and *n* the number of data points, with *n>K*, agglomerative HC starts from *n* clusters, then aggregates data until it obtains *K* clusters (example shown in Figure 3). Divisive HC, on the other hand, starts from just one cluster and then splits it depending, again, on similarities, until it obtains *K* clusters (example shown in Figure 4). When we talk about similarities, we are referring to the distance among data points, which can be computed in different ways.

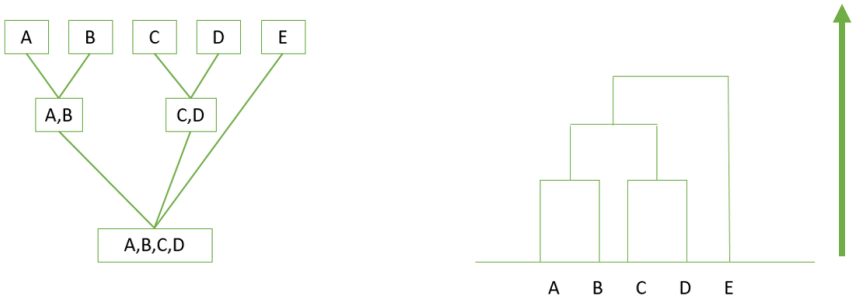


*Figure 3. Example representation of agglomerative Hierarchical Clustering.*

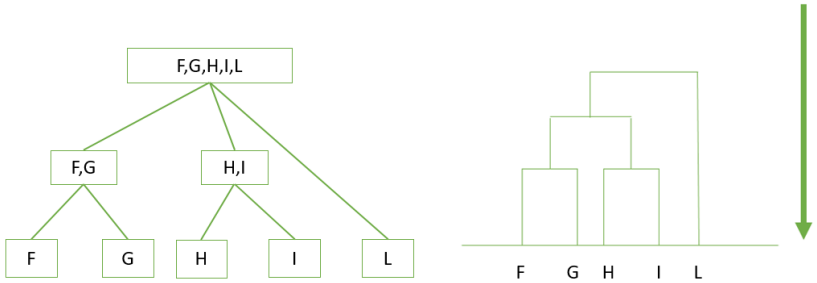


*Figure 4. Example representation of divise Hierarchical Clustering.*

In mathematical terms, similarity mainly refers to distance, and it can be computed with different approaches. For example:

**Min:** it states that, given two clusters *C1* and *C2*, the similarity between them is equal to the minimum of similarity (translated: distance) between point *a* and *b*, such that *a* belongs to *C1* and *b* belongs to *C2*.

**Max:** it states that, given two clusters *C1* and *C2*, the similarity between them is equal to the maximum of similarity between point *a* and *b*, such that *a* belongs to *C1* and *b* belongs to *C2*.

**Average:** it takes all the pairs of points, compute their similarities and then calculate the average of the similarities. That is the similarity between the clusters *C1* and *C2*.

## 4. Assessment/Evaluation of the proposal

### a. Project design and goal

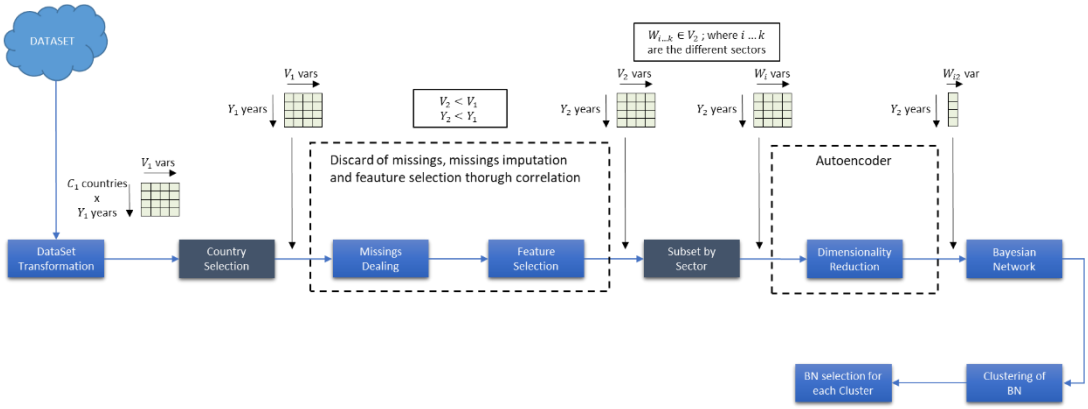


*Figure 5. Flow diagram of the global process.*

The goal of this project is to study the potential relationships among the aforementioned and subsequently described sectors. These sectors are the aggrupation of the variables contained in the dataset. They represent some of the different indicators that exist and describe a country or population (such as the economic growth, the health system, the defense system, education, technology, energy and other social and economic factors). What we want to analyze is if there exist any relationship among them. Besides, we do not only want to prove if those variables are just correlated but also if there exist a causal relationship between them. To accomplish this goal what we did was to use the Bayesian Networks to study if there exist those causal relationships and then use a clustering method to create different groups of countries (using the Bayesian Network obtained for each one as source of the clustering) and provide one Bayesian Network model for each of the clusters. In the following sections, we will go through all the steps that we have followed to accomplish our goal.

### b. The dataset

In this section, we will describe the original dataset and source used in this work. We were looking for a good-quality statistical data of countries indicators and we found "The World Bank". The World Bank Group is one of the world's largest sources of funding and knowledge for developing countries. They have five different institutions that share a commitment to reducing poverty, increasing shared prosperity, and promoting sustainable development. This is how they describe themselves:

*At the World Bank, the Development Data Group coordinates statistical and data work and maintains a number of macro, financial and sector databases. Working closely with the Bank's regions and Global Practices, the group is guided by professional standards in the collection, compilation and dissemination of data to ensure that all data users can have confidence in the quality and integrity of the data produced.*

*Much of the data comes from the statistical systems of member countries, and the quality of global data depends on how well these national systems perform. The World Bank works to help developing countries improve the capacity, efficiency and effectiveness of national statistical systems. Without better and more comprehensive national data, it is impossible to develop effective policies, monitor the implementation of poverty reduction strategies, or monitor progress towards global goals.*

A user can download the whole dataset for just one country or a set of them. We decided to use the data of the top 100 countries by GDP index. These datasets contain the following information for each country:

1. Time period: for each country there is data for almost the last 60 years, from 1960 to 2018. It is important to mention that not for all these years the completeness of the data is the same, for example the data from 1960 to 1969 is by far the one with the most number of missings.

2. The variables: there are almost 1600 variables that are grouped in 20 different sectors (the aforementioned indicators). The sectors are:
    a. Aid Effectiveness
    b. Agriculture & Rural Development
    c. Climate Change
    d. Economy & Growth
    e. Energy & Mining
    f. Education
    g. Environment
    h. External Debt
    i. Financial Sector
    j. Gender
    k. Health
    l. Infrastructure
    m. Private Sector
    n. Public Sector
    o. Poverty
    p. Science & Technology
    q. Social Development
    r. Social Protection & Labor
    s. Trade
    t. Urban Development

This separation in sectors will be crucial in this work, we will use them as the main descriptors of the countries and the correlation and causality descripted before will be carried out over these sectors. We will look for the relationships between these sectors and we will put the focus on the study of the possible causal relationships between them.

### c. Dataset Transformation

The second step was the data transformation, since the data structure was not very suitable for mining we had to develop a function to make it appropriate for the mining goal.

Figure 6 shows the structure of the original dataset where the columns were the time dimension (years) and the rows contained the different variables distributed and replicated for each country.

| Country Name | Country Code | Indicator Name | Indicator Code | 1960 | 1961 | 1962 | 1963 |
|---|---|---|---|---|---|---|---|
| Australia | AUS | Urban population (% of total) | SP.URB.TOTL.IN.ZS | 81,529 | 81,941 | 82,228 | 82,511 |
| Australia | AUS | Merchandise exports by the reporting economy (current US$) | TX.VAL.MRCH.WL.CD | 2022900000 | 2338500000 | 2322900000 | 2774300000 |
| Australia | AUS | Merchandise trade (% of GDP) | TG.VAL.TOTL.GD.ZS | 25,37022657 | 24,08005333 | 24,61709914 | 25,95414135 |
| Australia | AUS | Gross national expenditure (current US$) | NE.DAB.TOTL.CD | 18658304401 | 20154552581 | 19800649569 | 21711277859 |
| Australia | AUS | General government final consumption expenditure (% of GDP) | NE.CON.GOVT.ZS | 11,11312108 | 11,3372093 | 11,98896272 | 11,69852596 |
| Australia | AUS | Military expenditure (% of GDP) | MS.MIL.XPND.GD.ZS | 2,36954514 | 2,415139854 | 2,36369537 | 2,446634639 |
| • | • | • | • | • | • | • | • |
| • | • | • | • | • | • | • | • |
| • | • | • | • | • | • | • | • |
| Spain | ESP | Merchandise trade (% of GDP) | TG.VAL.TOTL.GD.ZS | 11,98353124 | 13,02320989 | 14,2929054 | 14,10141617 |

*Figure 6. Structure of the original dataset.*

Figure 7 shows the final dataset's structure after the transformation for proper mining, where columns now are the variables and the rows contain the time-geographical dimensions (country and year). Note that since we will study each country individually, the rows of each final sub-dataset will be just the time dimension (years).

| country | year | Obs_Indv | SP.RUR.TOTL.ZG | SP.POP.TOTL.FE.ZS | SP.POP.GROW | SE.PRM.ENRR | SE.PRM.ENRL.TC.ZS | SE.PRM.AGES |
|---|---|---|---|---|---|---|---|---|
| AUS | 1970 | AUS_1970 | 0,146790806 | -0,007528195 | 0,163132822 | -0,292071944 | 0,439154866 | -0,4115059 |
| AUS | 1971 | AUS_1971 | 0,377453196 | -0,001173609 | 0,994906168 | 0,756778989 | -0,071125714 | -0,4115059 |
| AUS | 1972 | AUS_1972 | 0,124947056 | 0,005720741 | 0,085245321 | 0,720963005 | -0,171300563 | -0,4115059 |
| AUS | 1973 | AUS_1973 | 0,074830406 | 0,013682077 | -0,097217666 | 0,678994242 | -0,239101941 | -0,4115059 |
| AUS | 1974 | AUS_1974 | 0,236847287 | 0,023261653 | 0,494066378 | 0,368888014 | -0,453543686 | -0,4115059 |
| • | • | • | • | • | • | • | • | • |
| • | • | • | • | • | • | • | • | • |
| • | • | • | • | • | • | • | • | • |
| ESP | 2017 | ESP_2017 | -0,174165646 | 0,365887847 | -0,887495109 | 0,954339173 | -1,088499262 | -0,4115059 |

*Figure 7. Structure of the dataset after the transformation.*

### d. Country selection

Once we have the dataset built the following step is to select the country that we want to study or analyze. By this, we will subset the global dataset and we will be working from this point with one different dataset for each country.

### e. Data cleaning, dealing with missings and imputation methods.

This dataset has more than 1500 variables for each country and this information is provided for almost the last 60 years (since 1960). But, this brings a disadvantage and it is the important amount of missing values that are in the dataset.

Now is time to deal with this missing values, to do that, first thing we had to do was analyze the distribution of this missings.

But before, we want to point out that it is not easy take a decision about how to deal with missing values and there is no standard or procedure about how to behave and what are the decisions to make when you find missing values in your data. It is always dependent on your dataset and the problem you are dealing with. In our project, the goal is to find causality relationships among the sectors (aggrupation of variables) for

the different countries. To do that we had the information of those variables for the last 60 years. But, we do not need precisely one year or another, and the analysis won't change if we lack for one country a lot of information for some years (for example 1960, 1963 and 1975) and for another country some different years (for example 1962, 1967 and 1971). For this reason, we decided to accomplish the missings dealing in two different steps.

There are two ways of analyzing the missing values, by observations (rows) or by variables (columns). We decided to first look at the missingness by variables. Let us take the country Spain as example. What we found is that there were around 1078 variables, out of the total 1596, with more than 50% missing values. We decided to set the threshold of direct discard a variable in 50% of missingness, this means that we will keep those variables that have at least half of their data informed. To give a more illustrative picture of the amount of missings that we had at first, let us show an histogram of the number of missings by variables in Figure 8.

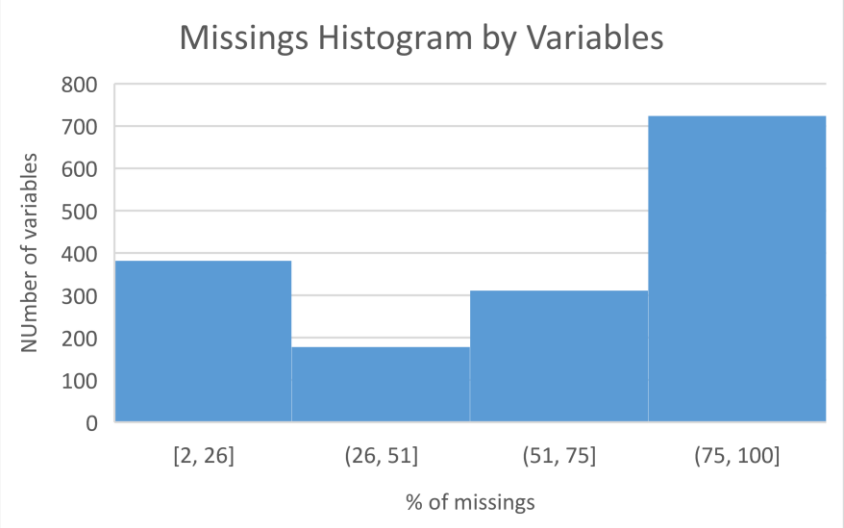


*Figure 8. Missings Histogram example.*

Subsequently, we had to analyze the missingness of the dataset resultant form the previous operation by observations (rows). We found that from the remaining data 10 years (10 observations, out of the total 59) still had more than the 50% of the data missing and those years where from 1960 to 1969. Not surprisingly, we discovered that when running the rest of the countries almost always this behavior was the same.

Now, the following task to complete our dataset is to impute the remaining missing values.

**Missings Imputation**

As we explained before, the step before applying a missing imputation algorithm is to identify the type of missing values that we have in our dataset. We pointed out that the type of missings that we have in a dataset depends on the data collection process. We are not the data collectors but we know that the source that we are using collects data from the different countries sources. Thus, we assume that when

the data is missing could have been for very different reasons due to the huge variety of variables that we have and the wide range of years and different countries that conform our dataset. Therefore, taking also into account that there is not a clear line separating the two first types of missings, described before, (MCAR and MAR) our missing could be a combination of them.

Consequently, among the different imputation methods that can be applied we decided to use KNN imputation because it is more efficient in terms of computational time and has reduced computing cost comparing to other more complex methods like MICE. It is true that using a more complex and powerful imputation method could improve the results in the way of having a more trustful dataset. However, for the moment, we decided that KNN imputation will be enough for this first attempt. Definitely, for a future work and new versions, invest more effort in the study and research of different imputation methods would be an improvement of the project.

### f. Dimensionality reduction and feature selection methods

Now we have a dataset without any missing value, but still there are some things we have to take care before starting the analysis of the data. We still have a huge dataset in terms of variables (around 500) and the first thing we had to analyze was the content of those variables, what they were talking about and if some of them pointed out to the same information. To do this, we made a correlation analysis of the variables, as we expected the reduction of the data was very significant (there were a lot of variables representing the same information but in different terms, for example *"Adjusted net national income (annual % growth)", "Adjusted net national income (constant 2010 US$)" and "Adjusted net national income (current US$)".*
Thus, it was not difficult to reduce a lot the number of variables with correlation analysis. In fact, the reduction was around ten times the number of variables, remaining by this way around 50 variables.

We decided to remove those variables with more than 80% of correlation. Here it happens something similar with missing analysis and it is that there is no standard or procedure to decide which percentage of correlation is suitable and enough to decide to remove these variables. Therefore, we assumed that if we had for example two different variables with at least 80% of correlation is enough to decide to just take one of them and remove the other one. The decision of which of them will be remove is totally at random since it does not matter to us which of them remains, the information kept will be the same.

### g. Sub-setting by sectors

After correlation analysis, the following part of our process is sub-set the data by the aforementioned different sectors. We will work with many sub-datasets as sectors we have. The size of each dataset will be the number of variables related to that sector times the number of observations that we already had (number of years for each country).

### h. Second step of dimensionality reduction: PCA vs Autoencoders

At this point, we have one data set for each sector. Now it is time to describe the two different solutions that we have worked with to accomplish the task of compressing all the variables of each sector into one, trying always to keep as much information as we are capable.

Firstly, we used Principal Component Analysis. As we mentioned before, PCA is one of the best linear techniques for dimension reduction. PCA performs dimension reduction by projecting the original *n-dimensional* data onto the $m < n$ dimensional linear subspace. Thus, PCA builds a global linear model of the data (an *m-dimensional* hyperplane) and the representation of the whole dataset is more accurate as closer is the value of $m$ to $n$.

Our following approach is to use a neural network as an encoder of information. As we explained before, an autoencoder is an unsupervised artificial neural network that learns how to efficiently compress and encode data then learns how to reconstruct the data back from the reduced encoded representation to a representation that is as close to the original input as possible. Autoencoder, by design, reduces data dimensions by learning how to ignore the noise in the data.

Therefore, we encoded all the remaining variables for each sector with PCA and with an autoencoder. In Figure 9 we present the comparison of the error (the task of validation was done by reconstructing the original dataset with both methodologies, for PCA using one dimension and with Autoencoder using one neuron in the middle layer) of the reconstructions for one study case (country Spain) for each sector with PCA and with an autoencoder.

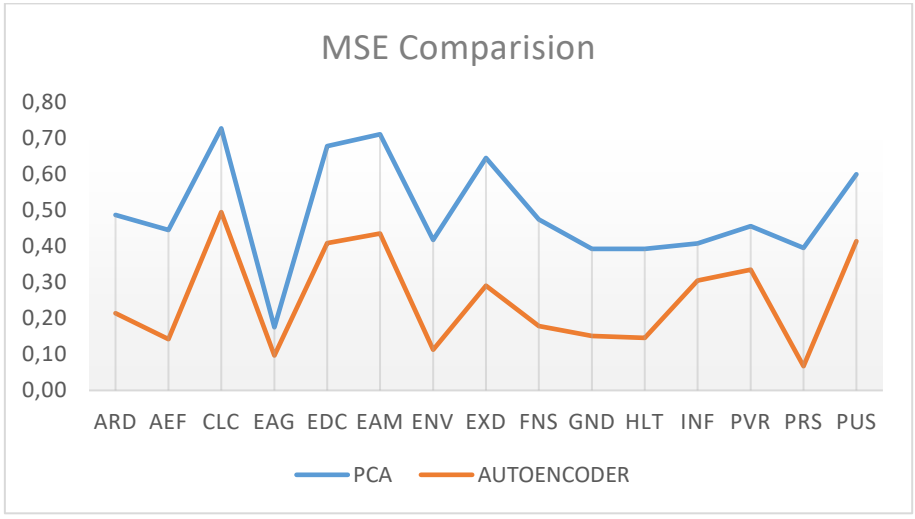


*Figure 9. Comparison of the reconstruction error for PCA and Autoencoder measured with Mean Squared Error.*

As we expected, PCA does not behave so well when we reduce to one the dimensions used for reconstruction. That is why we wanted to find another solution and as it can be clearly observed the results with an autoencoder are much better.

Therefore, even if the autoencoder is much more complex and less efficient in terms of computational cost, we will end up choosing this last option in order to have a better representation of the data.

### i. Bayesian networks

Heretofore, we have one sub-datset for each sector reduced to one variable as the output of the autoencoder. To briefly summarize, our dataset up to this point is composed by one variable describing and representing each sector and each observation is the data for each of the aforementioned years. Now, we are ready to build the Bayesian network. As we explained before, Bayesian networks (BNs) are a type of graphical model that encode the conditional probability between different learning variables in a directed acyclic graph. But before, building it there are some things to take into account regarding mainly to the distribution of the input data of the Bayesian Network. Before build a Bayesian Network, it is interesting to check if the data is normally distributed and if the dependencies among the variables are linear. So now the questions that arises are:

- **Is our input data normally distributed?**

  Let us analyze some of the variables' distribution for one example country (Spain). Figures 10 to 13 contain the histograms and QQplots for the variables "Climate Change" and "Economy & Growth" taken for this example. In the Table 1 it is shown the p-value obtained from the Shapiro-Wilk normality test for those variables.

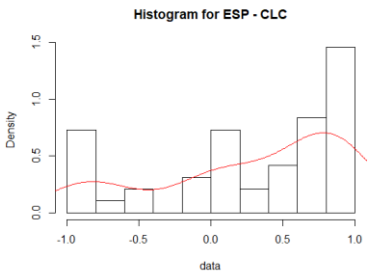


Figure 10. Histogram for country Spain and the variable Climate Change

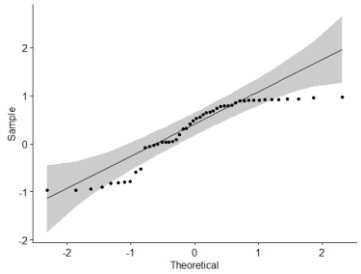


Figure 11. QQPlot for country Spain and the variable Climate Change

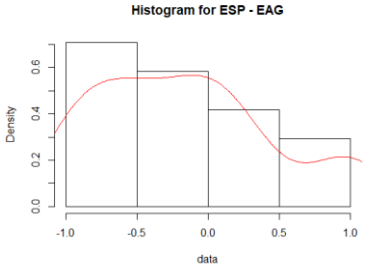


Figure 12. Histogram for country Spain and the variable Economy & Growth.

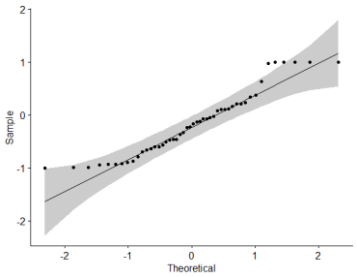


Figure 13. QQPlot for country Spain and the variable Economy & Growth.

| Country | Variable | P-Value |
|---|---|---|
| ESP | Climate Change | 0,0000382 |
| ESP | Economy & Growth | 0,00463 |

Table 1. Results from Shapiro-Wilk normality test.

These are just some samples of the global dataset, but the behavior for the rest of the countries and variables is the same, which is that the data does not follow a normal distribution.

- **Are the dependencies among the variables linear?**

For this task we performed an analysis based on a Scatter Plot between some of the variables. Following the previous example, the Figures 14 and 15 illustrates the Scatter Plots among 2 different pairs of variables for the example case Spain.

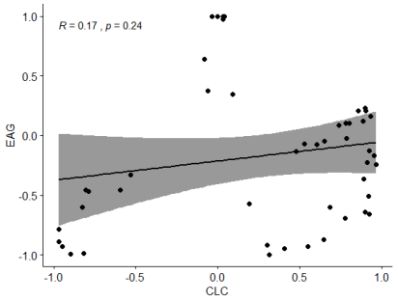


Figure 14. Scatter Plot for country Spain and the variables Climate change and Economy & Growth.

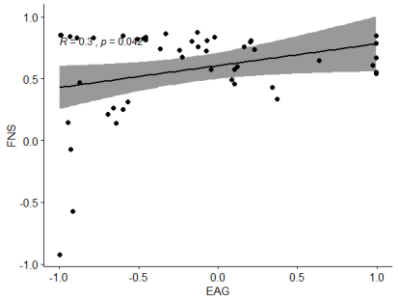


Figure 15. Scatter Plot for country Spain and the variables Financial Sector and Economy & Growth.

Therefore, we can as well as before, observe that the dependencies between variables are not linear.

- **What to do now?**

The solution that we accomplished was to discretize the data and to model the resulting dataset with a Discrete Bayesian Network (DBN), which can accommodate skewness and nonlinear relationships at the cost of potentially losing the ordering information. When discretizing data, one variable to be chosen is the amount of levels and there is not an easy or standard way to do this. We decided to use 3 levels representing concentration levels of the different variables (low, average and high concentrations). We used the Hartemink's Information-Preserving Discretisation method for discretization[19].

Now we are ready to develop the DBN. We used a scored-based algorithm, where each candidate DAG is assigned a score reflecting its goodness of fit, which is then taken as

an objective function to maximize. The algorithm used is a hill climbing greedy search that explores the space of the directed acyclic graphs by single-arc addition, removal and reversals, with random restarts to avoid local optima. It is important to mention that in order to be more robust, instead of just applying one time the mentioned algorithm, we decided to use a bootstrap solution. The used method estimates the strength of each arc as its empirical frequency over a set of networks learned from bootstrap samples. It computes the probability of each arc (modulo its direction) and the probabilities of each arc's directions conditional on the arc being present in the graph (in either direction).

This is how we get a Bayesian Network for each country in our dataset. In the Figures 16 to 19 we show some examples of the Bayesian Networks that we got from this experiment.

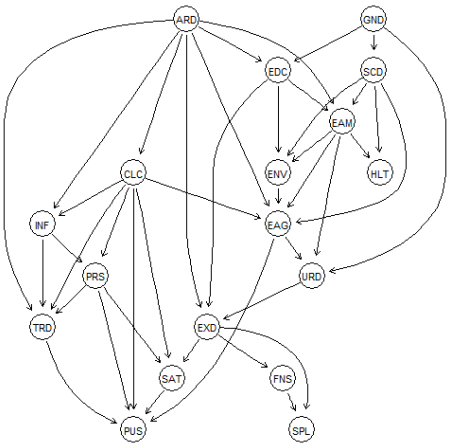


Figure 16. Bayesian Network representation of the country Spain.

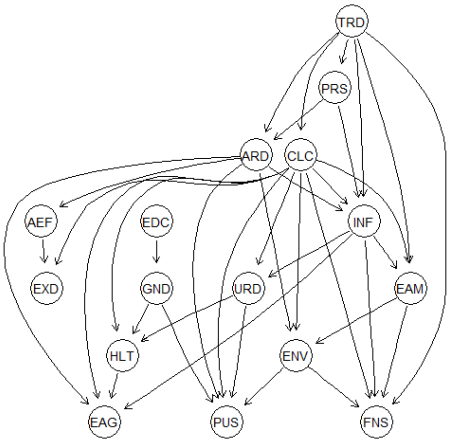


Figure 17. Bayesian Network representation of the country United Arab Emirates.

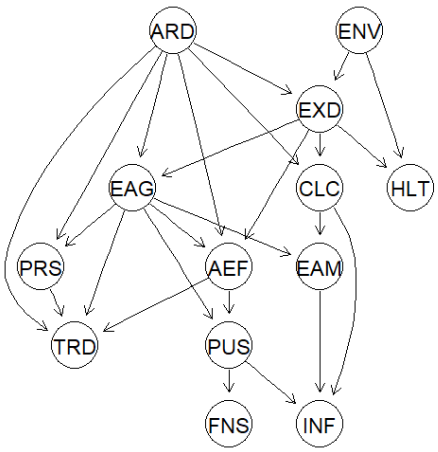


Figure 18. Bayesian Network representation of the country India.

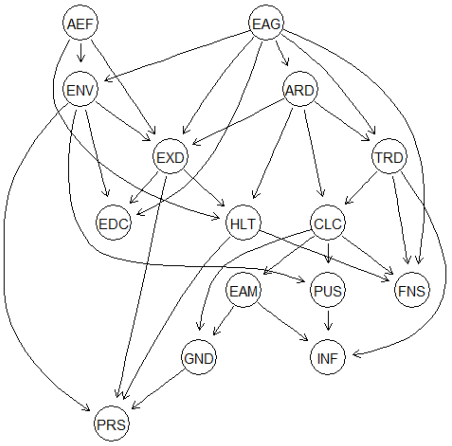


Figure 19. Bayesian Network representation of the country Iran.

## j. Clustering

Up to this point, we have generated one Bayesian Network for each country of our dataset, which is a graphical description of the causal relationships between the sectors or indicators of the country. The following step that we wanted to carry out was an attempt to group in several clusters those countries with similar causal relationships among their variables, which is those countries with similar Bayesian Networks.

The goal is to find *n* clusters of countries in the world that allow us to describe the *n* different systems that we can find in the different societies or populations around the world and try to give in this way a global picture of the world's most important architectures of functioning and socio-economic relationships for the different countries aggrupation.

To do this we first deploy an algorithm to represent a single Bayesian Network with a binary vector of factors. We first create a matrix where the rows and columns are the different sectors. Then the algorithm runs row by row and sets a value of '1' in each column if the selected pair row sector – column sector has a relationship father – child. This will give as a result a matrix as the one described in the Figure 20.

| | AEF | ARD | CLC | EAG | EAM | EDC | ENV | EXD | FNS | GND | HLT | INF | PRS | PUS | PVR | SAT | SCD | SPL | TRD | URD |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **AEF** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **ARD** | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| **CLC** | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 0 | 0 | 1 | 0 |
| **EAG** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 |
| **EAM** | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| **EDC** | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **ENV** | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **EXD** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 |
| **FNS** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| **GND** | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 |
| **HLT** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **INF** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| **PRS** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 |
| **PUS** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **PVR** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **SAT** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| **SCD** | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **SPL** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **TRD** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| **URD** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Figure 20. Adjacency matrix for country Spain. For each row it has a value of '1' in those columns that belongs to a sector which is a child in the Bayesian Network of the sector related to the selected row.

Figure 20, represents the relationships matrix for the country Spain. For each row we will find a value of '1' in those columns that belong to a sector that is a child in the BN of the sector related to the selected row. As consequence of that, if we look at one sector column, we will find a value of '1' in those rows that are parents in the BN of it. For example, the sector Energy & Mining (EAM) have the following relationships:

a) Childs: Economy & Growth (EAG), Environment (ENV), Health (HLT) and Urban Development (URD).
b) Parents: Agriculture & Rural Development (ARD), Education (EDC) and Social Development (SDC).

Once we have each country Bayesian Network represented in one relationship matrix, what we did was to convert them into vectors and generate a dataset where each observation (row) is the mentioned vector. This dataset is shown in the Figure 21.

| | V1 | V2 | V3 | V4 | V5 | V6 | V7 | V8 | V9 | V10 | V11 | V12 | V13 | V14 | V15 | V16 | V17 | V18 | V19 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **AUS** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **BRA** | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 |
| **CAN** | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| **CHN** | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| **DEU** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **ESP** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **FRA** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **GBR** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● |
| ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● |
| ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● |
| **URY** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **YEM** | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Figure 21. Relationships matrix generated by appending the relationships vectors of each country.

Therefore, now we have a dataset with all the representations of the Bayesian Networks of each country. This way, we can now clusterize this data. We want to point out that our data now is categorical and not numerical. On the contrary than clustering using numerical variables where the documentation and methods developed are abundant, it took us some time to find solutions for categorical data. It is due to the fact that categorical data clustering is less straightforward than numerical clustering and the methods for this task are still being developed.

Our clustering process include the following steps:

a) Calculating distance or dissimilarity matrix.
b) Choosing the clustering method.
c) Assessing clusters.

The first step is to calculate the dissimilarity matrix, which is a mathematical expression of how different, or distant, the points in a data set are from each other, so we can later group the closest ones together or separate the furthest ones. As for numerical data we have algorithms like Euclidean distance, for categorical data the most used one is the Gower distance.

To calculate the Gower distance we used the function daisy from R. Let us copy a fragment from R documentation to try to understand how the Gower distance works:

*Compared to dist whose input must be numeric variables, the main feature of daisy is its ability to handle other variable types as well (e.g. nominal, ordinal, (a)symmetric binary) even when different types occur in the same data set.*

*The handling of nominal, ordinal, and (a)symmetric binary data is achieved by using the general dissimilarity coefficient of Gower (1971). If $x$ contains any columns of these data-types, both arguments metric and stand will be ignored and Gower's coefficient will be used as the metric. This can also be activated for purely numeric data by metric = "gower". With that, each variable (column) is first standardized by dividing each entry by the range of the corresponding variable, after subtracting the minimum value; consequently the rescaled variable has range [0,1], exactly.*

*In the daisy algorithm, missing values in a row of $x$ are not included in the dissimilarities involving that row. There are two main cases,*

1. *If all variables are interval scaled (and metric is not "gower"), the metric is "euclidean", and $n_g$ is the number of columns in which neither row $i$ and $j$ have NAs, then the dissimilarity $d(i,j)$ returned is $\sqrt{p/n_g}$ $(p = ncol(x))$ times the Euclidean distance between the two vectors of length $n_g$ shortened to exclude NAs. The rule is similar for the "manhattan" metric, except that the coefficient is $p/n_g$. If $n_g = 0$, the dissimilarity is NA.*

2. *When some variables have a type other than interval scaled, or if metric = "gower" is specified, the dissimilarity between two rows is the weighted mean of the contributions of each variable. Specifically,*

$$d_{ij} = d(i,j) = \frac{\sum_{k=1}^{p} w_k \delta_{ij}^k d_{ij}^k}{\sum_{k=1}^{p} w_k \delta_{ij}^k}$$

   *In other words, $d_{ij}$ is a weighted mean of $d_{ij}^k$ with weights $w_k \delta_{ij}^k$, where $w_k = weigths[k]$, $\delta_{ij}^k$ is 0 or 1, and $d_{ij}^k$, the k-th variable contribution to the total distance, is a distance between $x[i,k]$ and $x[j,k]$, see below.*

*The 0-1 weight $\delta_{ij}^k$ becomes zero when the variable $x[,k]$ is missing in either or both rows (i and j), or when the variable is asymmetric binary and both values are zero. In all other situations it is 1.*

*The contribution $d_{ij}^k$ of a nominal or binary variable to the total dissimilarity is 0 if both values are equal, 1 otherwise. The contribution of other variables is the absolute difference of both values, divided by the total range of that variable. Note that "standard scoring" is applied to ordinal variables, i.e., they are replaced by their integer codes 1:K. Note that this is not the same as using their ranks (since there typically are ties).*

*As the individual contributions $d_{ij}^k$ are in [0,1], the dissimilarity $d_{ij}$ will remain in this range. If all weights $w_k \delta_{ij}^k$ are zero, the dissimilarity is set to NA.*

Thus, after calculating the Gower distance on our dataset we got the distance matrix showed in the Figure 22.

| | AUS | BRA | CAN | CHN | DEU | ESP | FRA | GBR | IDN | IND | IRN | ITA | ● | ● | SAU |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| AUS | 0 | 0,1275 | 0,13 | 0,1375 | 0,1275 | 0,1425 | 0,1525 | 0,155 | 0,13 | 0,125 | 0,14 | 0,15 | ● | ● | 0,1275 |
| BRA | 0,1275 | 0 | 0,1425 | 0,14 | 0,125 | 0,15 | 0,13 | 0,1525 | 0,1125 | 0,0975 | 0,1275 | 0,1475 | ● | ● | 0,12 |
| CAN | 0,13 | 0,1425 | 0 | 0,1425 | 0,1325 | 0,1475 | 0,1675 | 0,145 | 0,14 | 0,125 | 0,14 | 0,165 | ● | ● | 0,1275 |
| CHN | 0,1375 | 0,14 | 0,1425 | 0 | 0,155 | 0,16 | 0,175 | 0,1625 | 0,1325 | 0,1175 | 0,1375 | 0,1675 | ● | ● | 0,13 |
| DEU | 0,1275 | 0,125 | 0,1325 | 0,155 | 0 | 0,14 | 0,12 | 0,1575 | 0,1325 | 0,0975 | 0,1375 | 0,1475 | ● | ● | 0,125 |
| ESP | 0,1425 | 0,15 | 0,1475 | 0,16 | 0,14 | 0 | 0,165 | 0,1725 | 0,1625 | 0,1325 | 0,1675 | 0,1475 | ● | ● | 0,135 |
| FRA | 0,1525 | 0,13 | 0,1675 | 0,175 | 0,12 | 0,165 | 0 | 0,1825 | 0,1475 | 0,1375 | 0,1475 | 0,1675 | ● | ● | 0,145 |
| GBR | 0,155 | 0,1525 | 0,145 | 0,1625 | 0,1575 | 0,1725 | 0,1825 | 0 | 0,155 | 0,145 | 0,165 | 0,155 | ● | ● | 0,1325 |
| IDN | 0,13 | 0,1125 | 0,14 | 0,1325 | 0,1325 | 0,1625 | 0,1475 | 0,155 | 0 | 0,11 | 0,125 | 0,15 | ● | ● | 0,1175 |
| IND | 0,125 | 0,0975 | 0,125 | 0,1175 | 0,0975 | 0,1325 | 0,1375 | 0,145 | 0,11 | 0 | 0,1 | 0,14 | ● | ● | 0,0925 |
| IRN | 0,14 | 0,1275 | 0,14 | 0,1375 | 0,1375 | 0,1675 | 0,1475 | 0,165 | 0,125 | 0,1 | 0 | 0,165 | ● | ● | 0,1225 |
| ITA | 0,15 | 0,1475 | 0,165 | 0,1675 | 0,1475 | 0,1475 | 0,1675 | 0,155 | 0,15 | 0,14 | 0,165 | 0 | ● | ● | 0,1425 |
| JPN | 0,15 | 0,1475 | 0,165 | 0,1725 | 0,1525 | 0,1725 | 0,1625 | 0,185 | 0,16 | 0,15 | 0,165 | 0,175 | ● | ● | 0,1575 |
| KOR | 0,15 | 0,1225 | 0,14 | 0,1375 | 0,1175 | 0,1625 | 0,1225 | 0,15 | 0,13 | 0,11 | 0,12 | 0,165 | ● | ● | 0,1175 |
| ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | 0,12 |
| ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | 0,0775 |
| SAU | 0,1275 | 0,12 | 0,1275 | 0,13 | 0,125 | 0,135 | 0,145 | 0,1325 | 0,1175 | 0,0925 | 0,1225 | 0,1425 | 0,1575 | 0,1175 | 0,1 |

Figure 22. Dissimilarity matrix resulting from the calculation of the Gower distance over the relationship matrix.

Then, we go to the second step, choosing the clustering algorithm. We tried the two different methods from hierarchical clustering, agglomerative (bottom-up) and divisive (top-down).

Agglomerative clustering will start with n clusters, where n is the number of observations, assuming that each of them is its own separate cluster. Then the algorithm will try to find most similar data points and group them, so they start forming clusters. In contrast, divisive clustering will go the other way around, assuming all your n data points are one big cluster and dividing most dissimilar ones into separate groups.

To be honest, we made various trials with both algorithms and with different number of clusters and the results were very similar (we compare the results by calculating the correlation among the resulting dendograms), indeed, we could not say precisely that one of them was better or worse than the other one. We decided to carry out with the agglomerative (bottom-up) alternative (an illustration of the result is shown in the Figure 23).

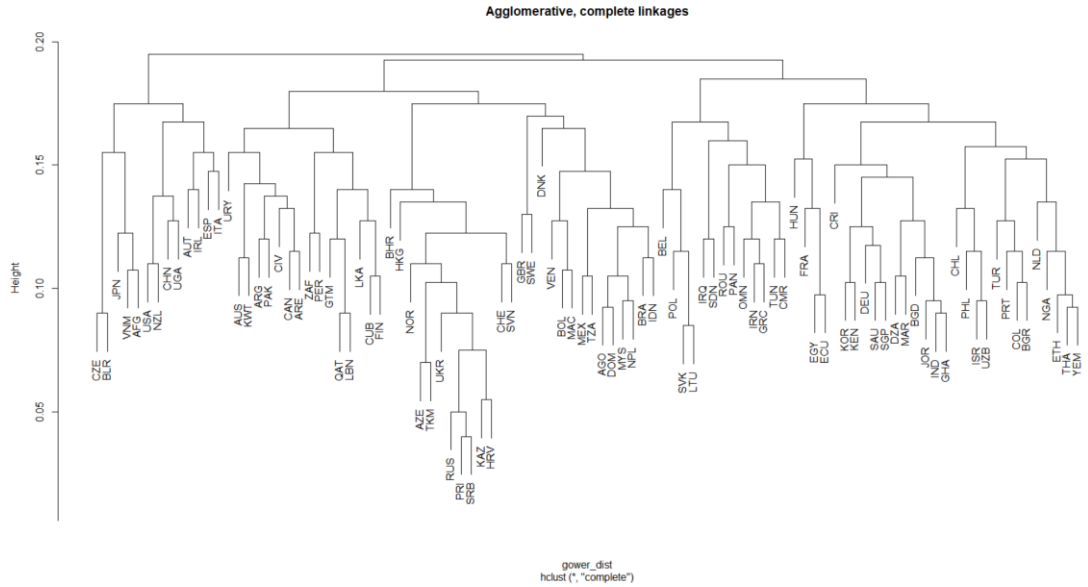


Figure 23. Illustration of the result for agglomerative clustering over our dataset.

Therefore, we just have to go through the last step, decide the number of clusters. This is not an easy and straightforward task and it depends a lot in the goal of the clustering, the data itself and our own judgement. Still, we used some algorithms that will help us in the task of assessing the number of clusters:

c) Elbow method: The elbow method is a heuristic method of interpretation and validation of consistency within cluster analysis designed to help find the appropriate number of clusters in a dataset. It is often ambiguous and not very reliable, and hence other approaches for determining the number of clusters such as the silhouette method are preferable.

d) Silhouette method: is a measure of data consistency, the silhouette plot displays a measure of how close each point in one cluster is to points in the neighboring clusters.

In the Figures 24 and 25 we show the results obtained after applying these methods.

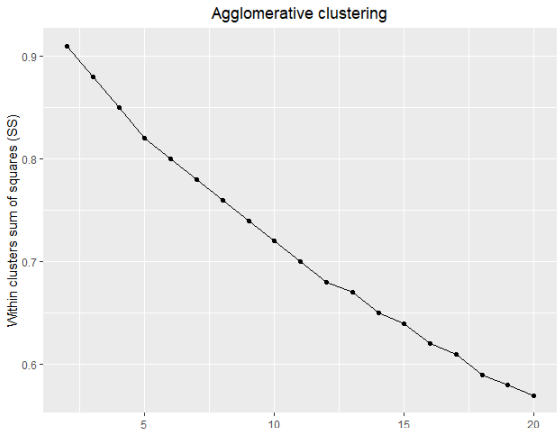


Figure 24. Elbow rule result.

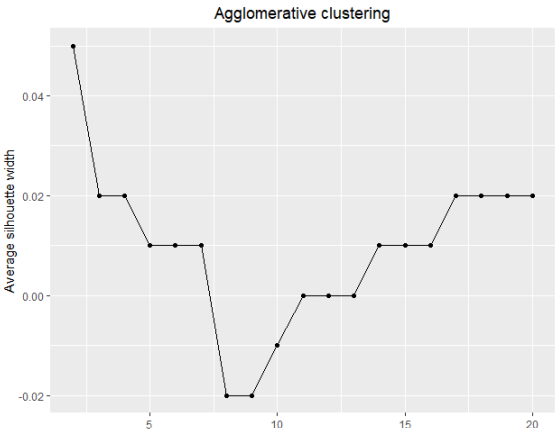


Figure 25. Silhouette method result.

As it can be observed, the Elbow method does not lead us to any clear conclusion. On the other hand, the Silhouette method gives an interesting result. According to it the best option would be 2 clusters, but is not for our interest just to clusterize all the countries in just 2 sets, then it start decreasing up to 9 clusters and increases then with the number of clusters. As we are interested in an amount of clusters between 4 and 10, according to this method we would chose 5 clusters.

Once we have decided the number of clusters, it is time to build the dendogram resulting from the clusterizaton, which is depicted in the Figure 26.

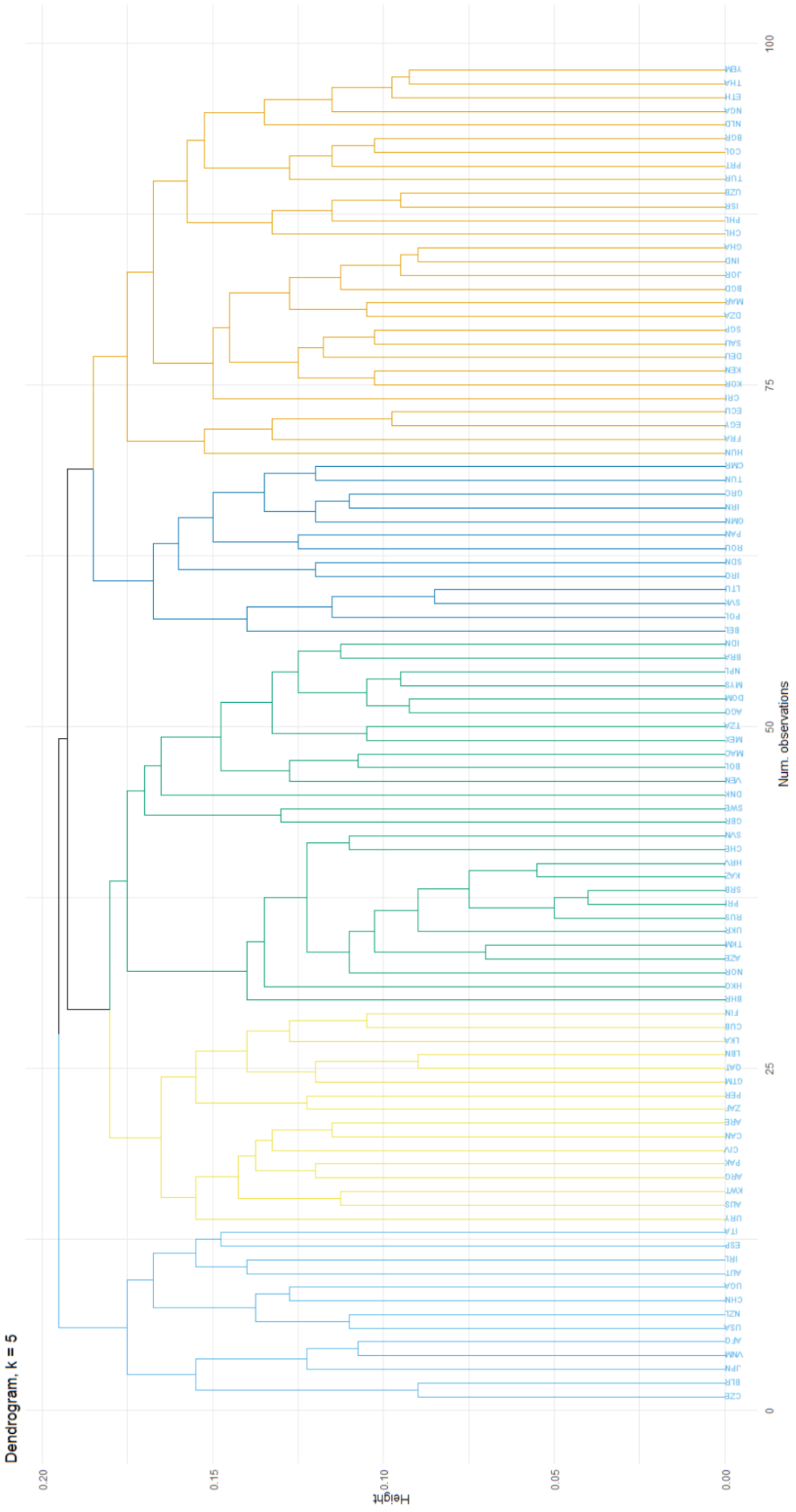


Figure 26. Resulting final dendogram after clusterize in 5 different clusters.

In the Table 2 we present the resulting clusters and its members.

| Country | Cluster | Country | Cluster | Country | Cluster | Country | Cluster | Country | Cluster |
|---|---|---|---|---|---|---|---|---|---|
| AUS | 1 | BRA | 2 | CHN | 3 | DEU | 4 | IRN | 5 |
| CAN | 1 | GBR | 2 | ESP | 3 | FRA | 4 | BEL | 5 |
| ARE | 1 | IDN | 2 | ITA | 3 | IND | 4 | IRQ | 5 |
| ARG | 1 | MEX | 2 | JPN | 3 | KOR | 4 | POL | 5 |
| PAK | 1 | RUS | 2 | USA | 3 | SAU | 4 | GRC | 5 |
| ZAF | 1 | CHE | 2 | VNM | 3 | THA | 4 | ROU | 5 |
| KWT | 1 | MYS | 2 | AUT | 3 | TUR | 4 | OMN | 5 |
| PER | 1 | SWE | 2 | CZE | 3 | BGD | 4 | SDN | 5 |
| QAT | 1 | DNK | 2 | IRL | 3 | COL | 4 | SVK | 5 |
| CUB | 1 | HKG | 2 | BLR | 3 | DZA | 4 | TUN | 5 |
| FIN | 1 | KAZ | 2 | NZL | 3 | EGY | 4 | CMR | 5 |
| GTM | 1 | NOR | 2 | AFG | 3 | NGA | 4 | LTU | 5 |
| LKA | 1 | UKR | 2 | UGA | 3 | NLD | 4 | PAN | 5 |
| CIV | 1 | VEN | 2 | | | PHL | 4 | | |
| LBN | 1 | AGO | 2 | | | SGP | 4 | | |
| URY | 1 | AZE | 2 | | | CHL | 4 | | |
| | | DOM | 2 | | | HUN | 4 | | |
| | | TZA | 2 | | | ISR | 4 | | |
| | | BHR | 2 | | | MAR | 4 | | |
| | | BOL | 2 | | | PRT | 4 | | |
| | | HRV | 2 | | | BGR | 4 | | |
| | | MAC | 2 | | | ECU | 4 | | |
| | | NPL | 2 | | | ETH | 4 | | |
| | | PRI | 2 | | | GHA | 4 | | |
| | | SRB | 2 | | | KEN | 4 | | |
| | | SVN | 2 | | | UZB | 4 | | |
| | | TKM | 2 | | | CRI | 4 | | |
| | | | | | | JOR | 4 | | |
| | | | | | | YEM | 4 | | |

Table 2. Resulting clusters and its members.

These are the clusters that we get as a result. There is not a standard or straight way to interpret this result, in fact, it is open to different interpretations. In the following step we will assign a Bayesian Network to each of these clusters and then we will try to give an analysis and extract some conclusions as final result of our project.

### k. Assigning one representative Bayesian Network for each cluster

The last task that we wanted to accomplish was to find a Bayesian Network that could perform as the representative network for each cluster. This was not an easy problem to solve and we are aware of the multiple possibilities and methods that could be used to carry out this task.

We came up with the following idea. As we explained in the previous section, the first step of clustering is to calculate the distance or dissimilarity matrix between all the countries of our dataset and therefore the final dendogram will be a clusterization that arises from those distances. So we fist split the global dissimilarity matrix in sub-matrixes, each one corresponding with each of the obtained clusters. Since, one country just can be placed in only one cluster this is a straightforward separation and we will have *N* (with *N* = number of clusters) different non-overlapping distance matrixes.

What we have now is the dissimilarity matrixes of each cluster, thus, we have the intra-cluster distances for each pair of neighbors. Therefore, if we sum the columns of the matrix for each row (since is a square matrix will be the same to sum the rows for each column) we will get the accumulative distance of each element of the cluster to his neighbors. For example, if there is a dissimilarity matrix of 3 elements:

| | ESP | FRA | DEU |
|---|---|---|---|
| **ESP** | 0 | 0,1 | 0,3 |
| **FRA** | 0,1 | 0 | 0,4 |
| **DEU** | 0,3 | 0,4 | 0 |

| Acc.Distance |
|---|
| 0,4 |
| 0,5 |
| 0,7 |

The accumulative distance of Spain to his 2 neighbors will be 0.4, the accumulative distance of France will be 0.5 and the accumulative distance of Germany will be 0.7.
Then we decided to take as the cluster representative Bayesian Network, the one corresponding to “the less distant neighbor”. In this case would have been Spain.

By this way, we finally have our final result, which are the final 5 clusters that will be a representation of the 5 different causal architectures that represent the different countries and populations that we find in our world. This results are shown in the Figures 27 to 31.

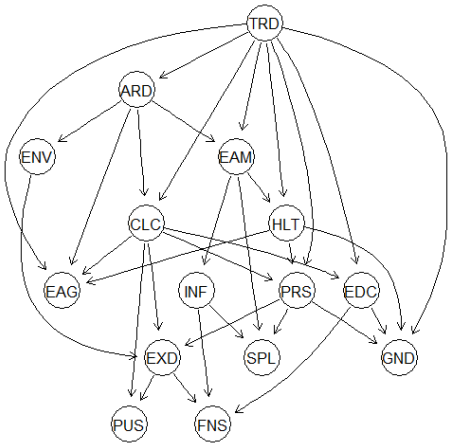


Figure 27. Cluster 1 representative (BN of country Finland).

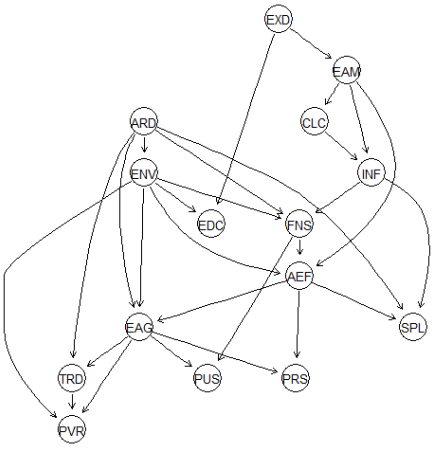


Figure 28. Cluster 2 representative (BN of country Brazil).

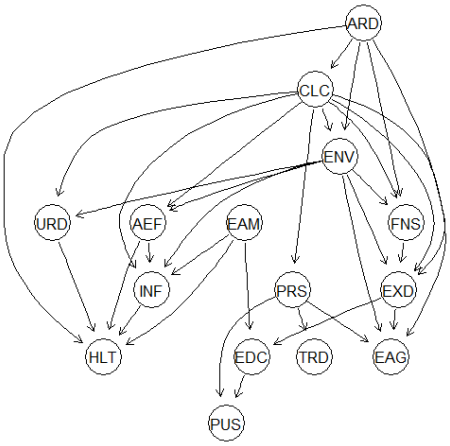


Figure 29. Cluster 3 representative (BN of country Vietnam).

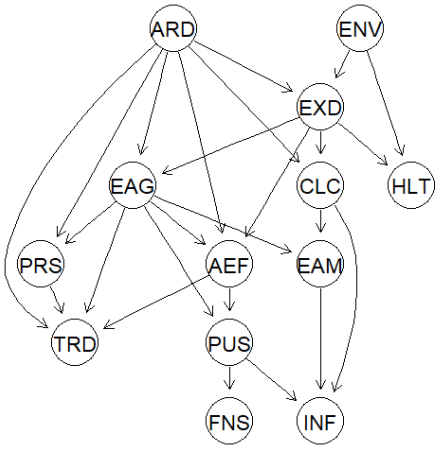


Figure 30. Cluster 4 representative (BN of country India).

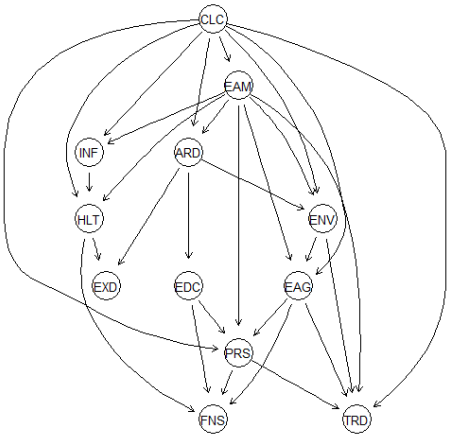


Figure 31. Cluster 5 representative (BN of country Slovak Republic).

These are the Bayesian Networks that describe the causal relationships of our defined 5 different clusters. Give a precise description of them is not an easy job, in fact, this kind of network is subjected to many different interpretations. Anyway, consecutively we are going to try to make a global interpretation of the final picture that we have shown before.

In general terms, we can observe that in all cases there are some different kind of nodes:

a) Those with many children and few or any parents, usually placed at the top of the graph (we will call them grand-father).
b) Those with many parents anf few or any child, usually placed at the bottom of the graph (we will call them grand-child).
c) Those with similar number of parents and children (we will call them standard-nodes).

Thus, we see some things that all the clusters have in common. For example, in all the cases the indicator "Agriculture & Rural Development" (ARD) is always a grand-father, this could have an easy interpretation and it is that the impact of the agriculture and the rural development along the last 60 years has been crucial for our countries and society.

Not leaving the analysis of grand-fathers, we can observe that in the Cluster 1 the indicator "Trade" (TRD) which has variables related to trading operations (like imports and exports of services, technology or weapons) is also playing an important role as grand-father meaning that many of the rest of the sectors have a causal dependence with it. This could be interpreted as that the countries of the Cluster 1 have a huge dependence on their trading (imports and exports) activity.

If we have a look at the indicator "Climate Change" (CLC) in the Clusters 1, 3 and 5 we can observe that it is also a grand-father in those cases. This could mean that for the countries of these Clusters the climate change is a very relevant indicator. Indeed, one interesting relationship is the one with the sector "Economy & Growth", which means that economical and developing factors (like the total income of a country, the savings, the GNI or GDP for example) have a direct causal dependence with the climate change (composed by many different variables like CO2 emissions, droughts, floods, extreme temperatures or amount of oil as energy source used).

Another interesting analysis is the position of the sector "Energy & Mining" (EAM) in the Cluster 5. Here, it has also a role of grand-father, and having a look at its children sectors we could interpret this as that the energy and mining activity have a huge influence or impact over the economy, society, private sector, health system and the infrastructures of those countries.

We can also observe that the indicators "Public Sector" (PUS) and "Private Sector" (PRS) are, in most of the cases, at the bottom of the network. Are two of the described grand-children. This could be interpreted as that the public and private sector (which are composed by variables like Cost of business start-up procedures, Time required to build a warehouse, Investment in transport with private participation [PRS] or Central government debt, Goods and services expense or Tax revenue and payments [PUS]) have an important dependence of all the rest of the things that are happening in a country (like the economy of the country, the finance state or the external debt).

These are some examples of possible analysis and interpretations of the obtained Bayesian Networks but as we mentioned before, it is not a closed analysis, on the contrary, it is opened to different interpretations.

## 5. Conclusions

After all this months of work (almost one year) the main conclusion that we have reached is that this project has complete a real data science challenge. First, we had an idea (Develop a data analysis of socio-economic and financial factors of different countries in the world) then we started gathering data from a real database with all the kind of issues that a real database brings with itself. Subsequently, there was a long travel of data analysis, facing problems, trying to find solutions, evaluate the different possibilities and making decisions.

Throughout this project, we have faced many different problems and all of them had different possible solutions or sometimes there even was not a solution itself. This was the real challenge, to make decisions. We had to take the decisions of how to deal with the different problems that we were finding in our way (missing values, high dimensionality, the dataset itself, and others) and we knew that these decisions could have an important impact in the final results. This is way we had to be very cautious and validate and challenge ourselves in every step that we decided to make.

Regarding to our results and taking the previous analysis into account, we can say that we have achieved our initial goal. The final outcome that we got is a reliable representation of the functioning of our world and each single country in particular. Also, we have succeeded in the task of giving as well a global vision of the dependencies between the different indicators that describe a country through a description of the causal relationships among them. Besides, we have additionally complete a clustering task with which we are able to group the analyzed countries in different sets according to the causal relationships among their indicators which can be interpreted as the socio-economic systems of these populations.

## 6. References


[1] World Bank. "*World Bank Open Data*" World Development Indicators, The World Bank Group, *https://data.worldbank.org*.

[2] Lorraine Li, "*Principal Component Analysis for Dimensionality Reduction*", Towards Data Science, May 24 - 2019, *https://towardsdatascience.com/principal-component-analysis-for-dimensionality-reduction-115a3d157bad*

[3] Nathan Hubens, "*Deep inside: Autoencoders*", Towards Data Science, Feb 25 - 2018, *https://towardsdatascience.com/deep-inside-autoencoders-7e41f319999f*

[4] Will Badr, "*Auto-Encoder: What Is It? And What Is It Used For? (Part 1)*", Towards Data Science, Apr 22 - 2019, *https://towardsdatascience.com/auto-encoder-what-is-it-and-what-is-it-used-for-part-1-3e5c6f017726*

[5] Wale Akinfaderin, "*Missing Data Conundrum: Exploration and Imputation Techniques*", IBM Watson Data, Sep 11 - 2017, *https://medium.com/ibm-data-science-experience/missing-data-conundrum-exploration-and-imputation-techniques-9f40abe0fd87*

[6] Ben-Gal, I. (2008). *Bayesian Networks*. In Encyclopedia of Statistics in Quality and Reliability (eds F. Ruggeri, R.S. Kenett and F.W. Faltin). doi:10.1002/9780470061572.eqr089

[7] Gámez, José & Mateo, Juan & Puerta, Jose. (2011). *Learning Bayesian networks by hill climbing: Efficient methods based on progressive restriction of the neighborhood*. Data Mining and Knowledge Discovery. 22. 106-148. 10.1007/s10618-010-0178-6.

[8] Hamed, "*Bayesian network in R: Introduction*", R-Bloggers, Feb 15 - 2015, *https://www.r-bloggers.com/bayesian-network-in-r-introduction/*

[9] Marco Scutari, "*Understanding Bayesian Networks*", Department of Statistics, University of Oxford, Jan 23 - 2017, *https://www.bnlearn.com/about/teaching/slides-bnshort.pdf*

[10] Marco Scutari (2010). *Learning Bayesian Networks with the bnlearn R Package*. Journal of Statistical Software, 35(3), 1-22. *http://www.jstatsoft.org/v35/i03/*.

[11] Valentina Alto, "*Unsupervised Learning: K-means vs Hierarchical Clustering*", Towards Data Science, Jul 08 - 2019, *https://towardsdatascience.com/unsupervised-learning-k-means-vs-hierarchical-clustering-5fe2da7c9554*

[12] Anastasia Reusova, "*Hierarchical Clustering on Categorical Data in R*", Towards Data Science, Apr 01 - 2018, *https://towardsdatascience.com/hierarchical-clustering-on-categorical-data-in-r-a27e578f2995*

[13] KAMBHATLA, Nandakishore; LEEN, Todd K. Dimension reduction by local principal component analysis. *Neural computation*, 1997, vol. 9, no 7, p. 1493-1516.

[14] JENSEN, Finn V., et al. *An introduction to Bayesian networks*. London: UCL press, 1996.

[15] PEARL, Judea. Bayesian networks. 2011.

[16] SAKURADA, Mayu; YAIRI, Takehisa. Anomaly detection using autoencoders with nonlinear dimensionality reduction. En *Proceedings of the MLSDA 2014 2nd Workshop on Machine Learning for Sensory Data Analysis*. ACM, 2014. p. 4.

[17] WANG, Yasi; YAO, Hongxun; ZHAO, Sicheng. Auto-encoder based dimensionality reduction. *Neurocomputing*, 2016, vol. 184, p. 232-242.

[18] R Core Team (2011). R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0, URL http: //www.R-project.org/.

[19] A. J. Hartemink. Principled Computational Methods for the Validation and Discovery of Genetic Regulatory Networks. PhD thesis, School of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, 2001.